CruiseControlling Your Project Source Code Repository

Supervision of the source code repository is a crucial element to ensure the success of any project. The author discuses a tool that helps you accomplish this task in a simple and efficient manner

Techknow Grid	
Language	Java
Platform	J2EE
Level	Intermediate

—Raja R.K

No matter what sophisticated source code version control tools you use, monitoring the changes made to your project source code repository is always vital to the success of your project. Monitoring becomes even more important when there are multi-site teams involved in project development. Even though most of the version control tools have some in-built notification mechanisms, they are either not feasible or time-consuming. In this article, we will show you how one could easily monitor alterations made to a source code repository using CruiseControl, a freely available tool.

CruiseControl is a framework for a continuous build process. It can be used as a tool to monitor changes made to the source code repository. The latest version supports a wide variety of plug-ins, including those for ClearCase, CVS, FileSystem, etc. You can get the complete list of modification set plug-ins from:

http://cruisecontrol.sourceforge.net/main/configxml.html#modificationset.

CruiseControl is free to use and you can download it from:

http://cruisecontrol.sourceforge.net/download.html. In order to use CruiseControl, you also need the Ant tool from Apache. You can download Ant from:

http://www.apache.org/dist/ant/binaries/

CruiseControl installation is fairly simple. First, extract the CruiseControl zip file on to your system hard disk. The downloaded file will not have the executable JAR file; you need to build it using the Ant tool. Edit build.xml under the main directory, remove the test targets (as we don't need them) and run the Ant tool. If everything goes well, the executable JAR file cruisecontrol.jar will be created under main/dist directory.

To run CruiseControl, all that you need to do is define two XML files – config.xml and build.xml – for your project and pass it to the CruiseControl tool.

A sample build.xml is shown in **code 1**. As we do not plan to use CruiseControl to build our source code repository, we define a dummy target build. CruiseControl will call this target whenever it detects a change in the source code repository.

Code 1

```
<project name="cc-helper" default="build"
basedir=".">
   <target name="build">
       <echo message="Nothing to build..."/>
   </target>
</project>
```

The next most important file is config.xml. This file should contain all the required configuration details for CruiseControl. Refer sample configuration file (**code 2**).

Code 2

```
<cruisecontrol>
    <project name="DeveloperIQ"
buildafterfailed="false">
        <modificationset requiremodification="true"
quietperiod="60">
     <ClearCase branch="developeriq-dev"
      viewpath="/vob/developeriq-dev">
     </ClearCase>
        </modificationset>

        <schedule interval="14400"> <!-- 4 hours -->
            <ant/>
        </schedule>

        <publishers>
            <htmlemail mailhost="email.com"
                returnaddress="rajark@email.com"
                skipusers="true"
                logdir="logs/developer_iq"
                xslfile="modifications.xsl"
        subjectprefix="File Modification Report - ">

                <always address="rajark@email.com"/>
                <failure
address="rajark@email.com"/>
            </htmlemail>
        </publishers>
    </project>
</cruisecontrol>
```

Note that the sample configuration file comprises three sections - modificationset, schedule and publishers. The modificationset is where you specify your source code repository details. In the sample file, we instruct CruiseControl to detect changes made to Clearcase VOB. Under ClearCase modificationset, you need to specify the branch and view path details. In the example, branch name is developeriq_dev and view path is /vob/developeriq_dev.

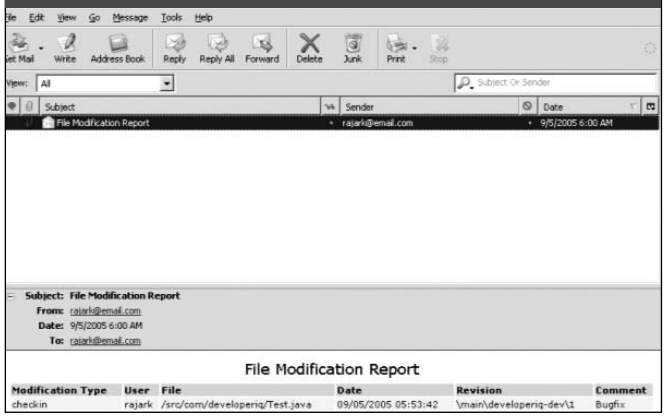
The schedule section is where you specify the monitor interval. In the example, we instruct CruiseControl to monitor ClearCase VOB for every 4 hours. Whenever CruiseControl detects changes to the source code repository, it will execute the instructions enclosed in Schedule section. In code 2, we instruct CruiseControl to run Ant tool, which will in turn (by default) call build.xml with the default target build.

Last, but not the least, is the publishers' section. This is where you specify the notification reporting details. In the example, we instruct CruiseControl to notify through Email Notification Mechanism.

Once you have defined build.xml and config.xml, you can start CruiseControl using the shell script cruisecontrol.sh under main/bin directory.

The following **figure** shows a sample email notification from CruiseControl.

Sample email notification



In the figure, you can see that the user rajark has checked-in a file /src/com/developeri/Test.java in \main\developeriq-dev\1 branch.

This sort of monitoring the source code repository not only helps the manager to handle and control the project source code but also helps the project team members to detect changes that could affect their module. DIQ

Reference

CruiseControl - http://cruisecontrol.sourceforge.net/
GUI for CruiseControl configuration -
GUI: http://cc-config.sourceforge.net/
Apache Ant tool - http://www.apache.org/dist/ant/

The author is working at HCL Technologies (Networking Product Division), Chennai. He can be contacted at: rajark_hcl@yahoo.co.in.